SUC-SESSION

- News
- ZX-Team meeting in Helmarshausen
- Retro-Programming with modern means
- A look under the rubber mat
- How the image gets onto the screen
- ZXPicoMD
- And much more...

ISSUE
235.
OF YOUR
MAGAZINE

YOUR MAGAZIN FOR THE SINCLAIR ZX SPECTRUM

EDITORIAL

Hello Spectrum friends,

There is a less pleasant reason, unfortunately, why I am allowed to write the foreword today. Our editorinchief and magazinemaker Joachim had an operation on his hand and is therefore simply unable to type. If all the bad luck comes together, then surely also during this summers silly season, in which already Joachim was largely absent as a writer, but also many others for professional and private reasons did not deliver as usual, hoped for or promised texts. At the moment I can't say what the final issue will look like, because Joachim can't work on it at the moment and will do so only slowly. Nevertheless, I am sure that he will put together a great issue.

Nevertheless, the magazine is by you and for you. In the end, it's not us editors who have to produce the magazine, but you readers yourselves. The contribution for the PDF issue of 4.50 Euros per year covers the running costs of the club, but unfortunately it is not enough for our SUC private yacht or jet, so we can't take over your work and do the whole issue on our own. In fact, we have done this very often in the past, in issue 233, for example, 8 of 14 articles were written by us, one translated by us and the foreword, letters to the editor and news in short were also largely written by us. In the end this is too much and we will not be able to do it again.

A lot has happened since then, the adventure corner has been handed over and the game reviews have been there for a long time anyway. But you won't find neither in this issue, because the texts are missing. That's not a bad thing, it's even good if more people realise the effort that goes into our texts. In addition, there is the layout of the texts in a booklet, the administration of the membership data, the printing and sending of the booklet or the PDF mails and the planning around it.

Don't get me wrong, we do that. There will always be articles from us, including in this issue. But essentially our job is to produce the magazine itself, not the articles. So if you want a full magazine, don't put it off until next week or even tomorrow. Start immediately. We need you. We need your articles, we need your feedback. Everybody can test a program here, everybody can present a new hardware they just bought anyway. Everyone can at least give us feedback, and in the end this can provide a piece of the puzzle that we can then turn into a whole magazine. Don't be afraid to send in a single program presentation either. Do you have a favourite game? Or even some rare hardware that hasn't been presented yet? It doesn't always have to be something new, especially we Spectrum users are always discovering "the old" anew. It's fascinating what has already existed, but has somehow disappeared again. There is a lot to write about here.

I'm sure we'll be flooded with articles now. It's a lot of work, but believe us, we're looking forward to it.

Have fun with this issue and special greetings to Joachim, who is now allowed to put this foreword into the issue.

Thomas

CONTENT

Editorial, Content, Imprint 2
How to subscribe to the SUC Session 3
Readers Letters 4
News in short 5
ZX-Team meeting in Helmarshausen 7
Retro-Programming with modern means - Part 2 . 8
A look under the rubber mat »Spectrum Inside« 9
How the image gets onto the screen 13
»The ZXPicoMD« 16
»Dizzy - 4 Magicland« 17
Keyboard membranes for retro computers 20

IMPRINT

The **SUC-Session magazine** appears in German and English and can be subscribed at the **»Spectrum-User-Club«**. For further information please contact **Thomas Eberle**.

Editor SUC-Session 234:
Joachim Geupel & Thomas Eberle

Editor SCENE+: Thomas Eberle

Club manager:
Mirko Seidel, Adresse: Birkenweg 8
89197 Weidenstetten
email: mirko.seidel1@gmail.com

This issue has been supported by:

Joachim Geupel	Joa.Geu@gmx.de
Thomas Eberle	thomas.eberle@sintech-shop.de
Harald Lack	
Heinz Junek	
Christof Odenthal	

Cover image by *»ZX Art«*

The rights of the articles as well as the pictures and graphics are owned by the individual authors and photographers.

We thank all helpers for their contributions.

Copy deadline for SUC SESSION 236
Sunday, 5. November 2023

HOW TO SUBSCRIBE TO THE SUC SESSION

In general, the magazine is the main service of the SPECTRUM-USER-CLUB. The magazine is made by Spectrum users and for Spectrum users. Everyone can participate. By participating you can earn the magazine.

The magazine is offered in subscriptions â three issues. Usually three issues are published per year.

Subscription 1:
Three issues as a printed version
Prices:
Germany: 22.00 €
EU-wide: 26.00 €
Worldwide: 29,00 €

Subscription 2:
Three issues as PDF
Price worldwide: 4,50 €

We are planning an exclusive access to reader-only content, software and older (backup) PDF issues of the SUC Session for download. Until the website for this is up and running, this content will be sent by email on request.

The club has the following bank details,

(only use this data, not the data of the Sintech company!):

IBAN: DE93630500000719102,
BIC: SOLADES1ULM,
account holder: Mirko Seidel

PAYPAL payment is also possible:
https://www.paypal.me/Micro256

or via our website
www.speccy-scene.de

Of course you can also use other payment methods like money by registered mail, cheque or other. It would also be important to send us an e-mail or call us for the address details.

Instead of paying in Euros, you can also pay in SUC points, 1 point is worth 1 Euro. There are the following points:

1) Submit article 2 points, activated on Publication
2) Translate article German-English or English-German 1 point
3) Enquiries and small letters to the editor do not get any points for the time being, but whoever sends a letter to the editor that can be used as a real article, will of course also get one point for it.

If you are interested in helping out, please contact us. We distribute translation work and try to keep things in order so that two authors do not work on the same topic.

If you already have a subscription, we will contact you when it expires. In this case please respond immediately so that we can plan the quantities for the next issue in good time.

READERS LETTERS

Thank you as always for your feedback. It is immensely important for us to get hints from you about what was good or what we can do better. Please write your comments as soon as you think of them to: thomas.eberle@sintechshop.de

Simone Voltolini **from San Giorgio in Italy still can't believe it:**

No disc anymore? ☹

Hi Simone,

to be honest, the SCENE+ was really a part of the work here in the SUC editorial team that was particularly fun. You had to hack the programs to get them onto disk, also test them and then also convert them for the different disk systems. Most of the time I've spent programming in my life has probably been spent rewriting routines to make the floppy version work. Many great programs have been converted to disk with the help of others and this legacy should not be lost. Rather, we want to present the contents digitally for downloading. We are not there yet, but we will be.

Fortunately, a sign of life came from *Ellvis* in Berlin:

Hi Thomas! Thanks for the last magazine and your message, I know I should have got in touch sooner. The second half of 2022 was tough for me and I was completely burnt out at work. Then in January I got a bad health diagnosis so I stopped all my activities. Now things are looking a bit better, but I am about to change jobs. I have always enjoyed writing the playtestes, but I don't have the time at the moment. I will need some time to reorganise myself, but that will take time at the moment.

Best regards, Ellvis

Hello Ellvis,

Nice to hear from you. Of course, the gamereviews section is a very important one for us, as we have to catch up with the daily programs at some point. But there are more important things, first and foremost health. We wish you all the best on your road to recovery and look forward to hearing from you again. Whenever you want to test a game, you are welcome to do so. At this point, however, I would also like to call on others: Ellvis can't make it at the moment. It doesn't have to be an article, you are welcome to send us a single game test. Last issue 234 had some games from 2021 (I don't think all the good ones yet).

We received an email from *Roger Boesch* from Switzerland. Roger wants to get more involved in the magazine and write articles. He has some questions about this:

Hi Thomas,

I don't know exactly how you organise it, but for me personally it would be easier to write an article if I knew the "long" beforehand and if there was a theme.

For me, it would be better to have some kind of media plan (accessible Google Doc. or similar) for the coming months than a call for articles.

or similar) for the coming issues. (more plannable for me).

My main skills are in programming and I could imagine making a small series out of it.

But since I'm involved in a few projects, I usually need more lead time.

But since the "Sinclairs" are always a favourite of me, I would actually like to do this to actively help.

But to be concrete, what I could offer in terms of programming:

Developing a game in C for ZX Spectrum and/or Sinclair QL. Preferably as a series, so that in four issues a "simple" game is created.

Let me know if that would be something and by when you would have to have the first part?

Greetings, Roger

Hello Roger,

I understand what you mean, but we are not that professional. Members write articles as they can spare the time. Plans are thrown around, some don't even get in touch because of stress at work. Others simply cancel their participation altogether. But others join in. We can't and don't want to limit ourselves by setting any kind of schedule. Everyone can submit what they want, the magazine is by users for users. We have a few regular sections that we take care of ourselves. These are "News in short", "Games (software reviews)" and also a bit of the "Adventure Corner", which Miron has taken over. That's it, everything else is always changing and there is neither a plan nor guidelines.

The good thing about it is that we are always open to new things. No one is restricted, everyone can participate.

I personally would be very interested in your series of articles on programming, but I'm sure others would also find the topic exciting. I will be very happy to receive an article or series of articles from you. The deadline for the next issue is in 2 days, but I've already agreed with Joachim that we have to extend it by another month. So if you have something ready by the end of May, that would be great. If not, then it will come one issue later. We don't want to rush anyone here, but it would be a great thing.

Dr *Heinz Junek* from Potsdam has only been a SUC Session reader for a short time, but he wants to get involved, which we think is great.

Hi Thomas, so far I have always corresponded with Joachim on this matter. For the next issue I would like to offer two articles:1. the ZXPicoMD (1 DIN A4 page, in continuation of an article in SUC233). This article is already finished and is possibly at Joachim already. I

can send it again. 2. a programmable joystick for the ZX Spectrum (about 23 pages, but still needs to be written). If there is interest in this, please let me know.

Greetings, Heinz

Hello Heinz,

I don't think we can run enough on the subject of hardware. The Spectrum is and always will be a computer on which a lot of hardware is developed, probably also because Sinclair didn't deliver much and didn't exist for long. So we like to see any kind of interface in our magazine, whether as a building instruction or a presentation.

Hello dear readers,

I'm finally getting a word in the end.

Thomas already mentioned it in the foreword: I've had an exhausting time; a lot of frustration in the club where I train, sickness, (fortunately) holidays, hand surgery and, last but not least, pneumonia, which I admit I didn't take seriously at first.

Writing is still difficult because my wrist is currently stiff from the operation, but this is gradually improving. I'm also getting my breath back and, fortunately, Thomas took the preface off my hands because I couldn't hold back with my preface once again thanks to Thomas, besides I sat at the keyboard for four days typing it in before it disappeared into the wastebin.

And I'll solve the club issue after the summer holidays.

I promised to publish a list of the existing magazines of the Spectrum User Club Wuppertal SPCW and, as successor, those of the Spectrum User Club SUC in this issue. I would like to apologise for the fact that nothing came of it. I started it then the trouble in the club started and my motivation to do anything for anyone at all was all gone; with the exception of my family of course. For the same reason, there are no text contributions from me this time, apart from this one. I have prepared a few things in the runup to this issue that I will sprinkle into the next issues.

Thomas and I are going to withdraw somewhat from the editorial work. This does not mean that there will be nothing more from us, but it does mean that there will be less from us. From my side, the "RetroComputer" series is coming to an end. I have pretty much finished all the computers in my collection, with the exception of the 16bit computers. Of the

8bit computers, the Atari computers of the XL and

XL and XE series, the MSX computers, which are absolutely interesting, and the Commodore computers C64 and C128. Also missing is the Sinclair QL, which actually already belongs to the 16bit computers. I will not describe the Commodore machines, after all, everyone knows them, and for the Atari computers there are specialists who know them better. They are invited to present their favourite computers. I will also introduce the MSX computers and that's it.

The future will show how the magazine will continue. It is created anew from one issue to the next. In the end, it depends largely on your cooperation.

Greetings,

Joachim

NEWS IN SHORT

NEW NEWSLETTER

»On the Spectrum« is the name of a new newsletter that is planned to appear every 4 weeks and will be sent by email. The content will be about wellknown classics, but also games that have not achieved a high degree of popularity. Furthermore interviews with personalities around the Spectrum, reviews and also novelties. It costs nothing to register:

https://onthezxspectrum.eo.page/onthezxspectrum

CRASH COVERTAPE

The cover tapes are back. But this time they don't come with a magazine. Are they still cover tapes then? Doesn't matter. Anyway, Fusion already has two cover tapes on the market for Crash Magazine. Both are already sold out, but they might be reissued. It certainly helps if you register as an interested party. You can do this by clicking on the blue lettering below the "Sold out" message.

CRASH Cover Tape #1:

Mr. Hair's a very hairy remix, Mr Hair and the kitty katacombs (Demo) TJ's cavern capers, 1Bit Mechanistic Music demo

https://fusionretrobooks.com/collections/crashmagazine/products/crashcovertape1?variant=39723458691166

CRASH Cover Tape #2:

Mecki Winter Hibernation, Savage Pricness 2 Demos, Sum it Sup, Astro Phobos

https://fusionretrobooks.com/collections/crashmagazine/products/crashcovertape2?variant=39775412387934

FTP CLIENT FOR ZX SPECTRUM

With *»SARAH«* you can log in to a FTP and download Spectrum files (games). How the whole thing works can be seen in a video:

https://www.youtube.com/watch?v=ZMfyZwLZNL4

The problem is, of course, that the Spectrum has no Ethernet port and, as always on the Spectrum, there are already several standards that are not compatible with each other. One is the Spectranet, the Spectrum Next and this solution here is for the few who have an MB03+ (not MB02+). It is really very nice, but only a few will use it. Also, no MB03+ are being produced at the moment because of the chip shortage. But this might change again and then SARAH might become interesting for even more. Download:

https://hood.speccy.cz/dwnld/SARAH1.0.zip

TOWNSEND GAMES IN GERMAN

Clive Townsend has been quite busy lately, not only with games for the "Saboteur" character. Some, if not all, programs are currently being translated and also brought to market in German. The extent to which the market for this is very large remains to be seen, but we have already been able to look at some games and talk to the translator Daniel Roth. The project is already very far advanced, and we will report on how and when the games can be purchased.

THE NEXT SABOTEUR

Another secret from *Clive Townsend's* workshop: »Saboteur« will soon be released on NEXT. I was able to sneak into the workshop and take some photos with my mobile phone. The whole thing looks very fluid, the graphics well done. But it is recognisably the good old Saboteur game exactly like on the Spectrum. How exactly... we will find out when it is finished.

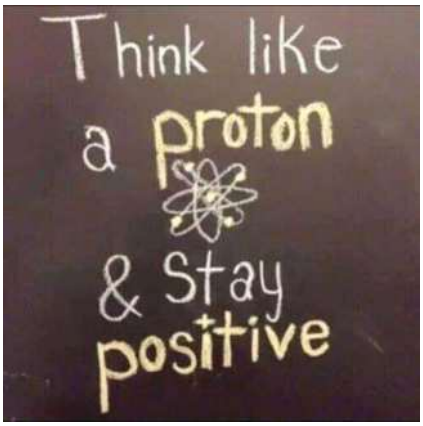
SPECTRUM EMULATES CHIP8

Thomas Tombinha from Hamburg is currently finishing an emulator for the Spectrum. So it is not the Spectrum that is emulated, but the Spectrum emulates another computer, in this case one of the first simple game consoles from the 70s: Chip8 and also Super-Chip8. We will of course report
on this in more detail after completion.

LOAD MAGAZINE, ISSUE 9/2023

The *»Verein zum Erhalt klassischer Computer (The society for the preservation of classic computers), VzEkC e.V.«* for short, has published its new club magazine. This time the focus is on "Operating Systems". The cover topics include several articles such as "The History of Operating Systems" and their application on different computer systems. Further, the computers of Atari, Apple and the (or possibly "the") inevitable Amiga are represented. The systems and company history of Tandy Radio Shack TRS80, WANG Laboratories, the history of the TA M32 computer and more are presented. Some of the articles are written by real insiders and are quite special as they deal with the characteristics of very special applications. Of course, the club news is not missing, as well as reports on meetings, such as the Regional Meeting Wittlich 2023, Retrocomputer Meeting Lower Saxony and the Classic Computing 2022.

The magazine is great and very informative. Some articles require specialist knowledge in certain subject areas, which does not diminish the reading pleasure. The magazine is worthwhile. (jg)



ZX-TEAM MEETING IN HELMARSHAUSEN

BY THOMAS EBERLE

This year the ><ZX-Team" met for the 26th time. The location had changed, the previous times it took place in Mahlerts in Hesse, this time it was in a youth hostel in Helmarshausen. I already visited a ZX-Team meeting about 20 years ago.

At that time, three of us travelled there and we were the only Spectrum users, while the ZX-Team consisted of ZX81 users. In the meantime, we know some of the ZX81 users better, because we meet more often, for example at the Spectrum meeting in Wittenberg. But also more Spectrum users visit the meeting more often and so the whole thing has rather turned into a Sinclair meeting, with ZX81, Spectrum and also QL.

This time too, three of us travelled, my son Noah, my long-time Spectrum friend Fred and me. Most of the visitors took advantage of the all-inclusive overnight offer of the youth hostel, while we were only day visitors. So we left early, but arrived in the deepest Hessian hinterland just before lunchtime. As picturesque as the landscape is, motorways are simply faster. But the location was impressive: a castle-like building, huge and several storeys high. After a few corridors, we found the room and were surprised that despite the large building, the room seemed a bit small. But this could also be due to the number of visitors, almost 60 people crowded into the room, there was no chance of a sales exhibition. With a lot of effort we managed to get half a table, after Dieter was kind enough to squeeze in a little. I had to set up my flea market, which I had brought with me, in the hallway, not a bad place, but I couldn't manage sales and sit at my Spectrum at the same time. It ended up with Fred and I going through some programs in the hallway with a laptop on our laps and Noah at the square trying out the latest SpectrumNEXT releases. This is not meant to be a criticism of the organiser, I just want to show the cramped conditions, but sometimes pictures speak louder than words: just look at the picture of the event.

Unfortunately, there were not too many lectures, but there was one on "Sinclair Computers in Movies", which was quite amusing and fortunately took place in another room (which I would have thought was bigger and therefore more suitable for the entire meeting).

The organisation was excellent, there were drinks in the room and three meals a day including drinks. Even if this sounds cheap at first, the costs were apportioned and amounted to 30,00 € per visitor (+ possible accommodation costs of 50-60 € per person). So those who are used to the low prices in Wittenberg may wince, but of course such a meeting is something unique and it is very well attended. In addition, the area around Kassel is pretty much in the middle of Germany, so it's easy to reach for everyone in the end, even if the journey from deepest Bavaria can still mean eight hours or more. In the end, there is hardly a better place to hold a multi-day meeting and I suspect a larger hall would have meant even higher costs, unfortunately. But it was unusual to see so many people together in such a small space, given the Corona pandemic we had just survived, and also the reason why some cancelled.

Because of the small space I can't say I saw much, but it was a pleasure to see such a full room with people who share the same hobby with me and what I have to change next time is to take an even bigger wallet and book the full programme, because the time was too short. I am already looking forward to the next time.

RETRO-PROGRAMMING WITH MODERN MEANS - PART 2

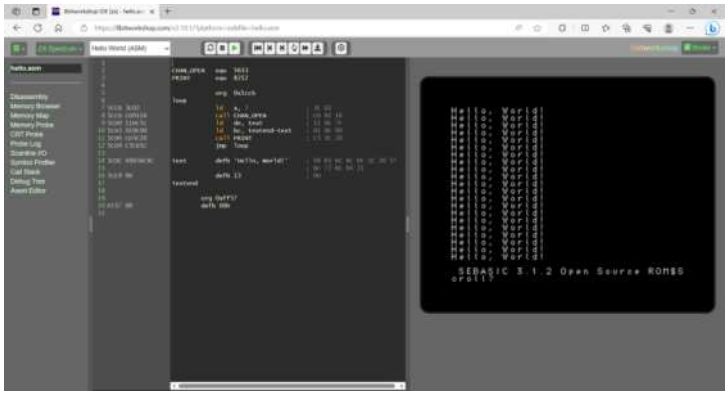
BY THOMAS EBERLE

Hello Spectrum friends,

After an article appeared in issue 233 about how to set up Visual Studio Code on your PC to program Z80 assembler with it, I did the same. In my personal opinion, but shared by some in my environment, Visual Studio Code is a good solution for those who are programmers anyway and who know how to set up and work with Visual Studio Code. For myself, my nerves failed me when setting it up and it took me a long time before the first short program was output as a .Z80 file. Once set up and understood, it's certainly a good solution, but I'll still show 2 alternatives here that my research revealed:

8BIT WORKSHOP IDE

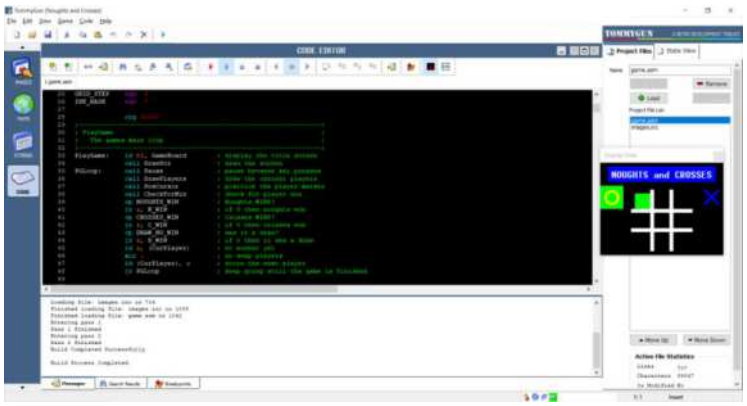
At **https://8bitworkshop.com/** you can find an online solution to program different systems in machine code. There are many devices to choose from, from Atari game consoles to C64 and CPC to the ZX Spectrum. I find the simultaneous emulation very good, so you can choose to program and see directly on the right side of the screen what has been programmed.



Unfortunately, this also reveals a weakness: I was not able to freely design my own routine. Instead, I could only modify the example routine "Hello World" shown here. As soon as I changed the ORG commands at the beginning or at the end, I no longer had a display on the right-hand side and thus could not test the correctness of my routine. Another minus point is the poor memory function. The routines are saved in the computer's cache, but if you delete the CACHE, everything would be gone. There is also the option to save the whole thing as a ZIP file, but this cannot be loaded back. There is still the option to copy and save the code using a text editor, but this is not a good solution. Perhaps there will be an update.

Despite these weaknesses, the website is suitable if you want to program something on the road. You only need internet access, and you could even develop routines on your mobile phone and then copy them later and use them in other software. So the whole thing is more of a tool, but not a whole development platform.

TOMMYGUN

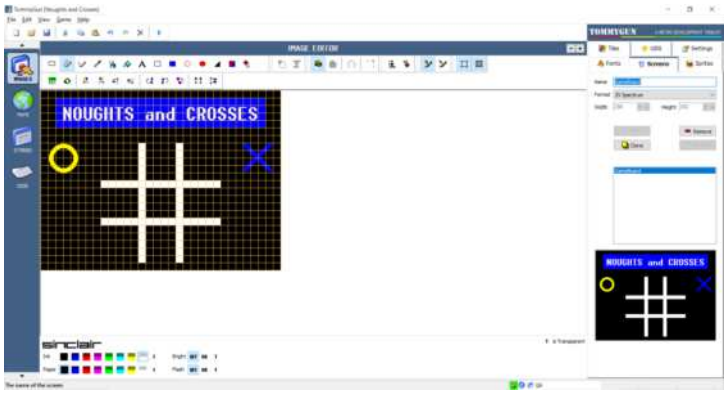


Tommygun is quite different, but also not fully developed. It works on Windows computers and also needs to be installed. Tommygun is a whole development platform, but in my opinion the developer (whose name is Tony, not Tommy) has taken on too much. There is even a special version that works especially with Jonathan Cauldwell's AGD (Arcade Game Designer). Furthermore, the program includes a section for graphics and one for coding. The whole thing is not necessarily designed for the Spectrum only. Tommygun only provides an interface, you specify yourself which assembler you want to work with and also which emulator. Tommygun is also known in the CPC area, but it is mainly used by Spectrum users and optimised here.

The graphics area can contain pictures or you can create sprites. I don't find working with the screen editor as comfortable as with graphics programs, but the sprite editor is quite well solved. Then you have the possibility to save the graphics in different formats and thus integrate them into the code. The programming section is nothing more than a text editor, the lines are only checked by the compiler when compiling. Here the use of PASMO is recommended, but also SJSAsm is possible (for assembler) or who wants to program in C chooses the Z88DK or in Basic with the Basic Compiler. As I said, Tommygun leaves a lot of freedom here.

It is possible to create small routines, save them and load them again. You create a main file and can then include the secondary routines in it (INCLUDE). The finished code can then be called up in a kind of test mode, in which case the created program is displayed in a small side window. Or you compile and create a TAP file, which can then be started directly in an emulator. This is then for the final test.

Strangely enough, the possibility of loading graphics is solved differently in the graphic editor than in the code editor. However, you can also insert graphics from another program by copy & paste. In this way, you can combine the topic of graphics with other programs for image processing (e.g. SEVENUP), since Tommygun gives you better possibilities to process the images into code (in the same format as the Spectrum screen layout or line by line). Elegantly, you can create



the graphics in such a way that when you change them in the editor, the code created from them also changes. If you don't want this, you only have to change the lines "Resources" in the code of the pictures.

Another negative point of Tommygun are the many strange messages and crashes that occur again and again. You can easily crash the program if you open a project, close it and open another one. So you have to close the whole program every time you want to look at one project and then switch to another.

Development of Tommygun ceased in 2014, although a small update was released in 2017. I think in the end Tony Thompson took on too much, a music program was also planned. In the end, that would have meant a graphics program, a music program and a programming environment, which would have been great if it worked, but what we have at the moment is not a music program, a painting program that lacks important functions and an editor that works so far and is the best thing about it. Maybe we should have concentrated on that, but maybe the development will continue at some point. You can get Tommygun, for example, at:

`https://sourceforge.net/projects/tommygun`

All in all, I have to say that I haven't found the ideal solution yet. But maybe you have one. Please write to the club about how you develop programs. Even if it is still done classically on the Spectrum, you can write about the advantages of this program development. We like to hear from you:

thomas.eberle@sintechshop.de

(te)

A LOOK UNDER THE RUBBER MAT

»SPECTRUM INSIDE«

RETROSPECTIVE: © 2023 HARALD LACK

Hardware used: Original ZX Spectrum 48K
Published in the Spectrum User Club

Dear 8Bit (Spectrum) friends!

The Sinclair ZX 80 as well as the ZX 81 and its direct successor, the ZX Spectrum, are well-known systems in 8-bit computer circles. Our Speccy has been around for more than 40 years and the developments around this machine by the fans are, in my opinion, more than unique. It takes a profound knowledge to adapt all these innovations to a computer system that is definitely getting on in years. What was new about these Sinclair computers some 40 years ago was Sir Clive's penchant for reducing components, which could almost be called ingenious. Here, tricks were used that even for us, as long-time Spectrum fans, are sometimes difficult to understand or remain forever incomprehensible. Not infrequently, the tricks used went far beyond the normal exploitation of existing components (especially the Zilog Z80 should be mentioned here, which was subject to quite "interesting" adaptations). Almost any (hidden or known) feature of any component, any signal present anywhere in the system was used in unusual and unforeseen ways to add needed functionality to the system. As a representative example, I would like to mention the type of image generation that resulted from an almost grotesque misuse of the interrupt. And then there was the mysterious ULA, a component that was not found in any other computer of that time. So far nothing new in Spectrum circles. And yet it remains somehow inexplicable to many of us, especially if you are not a technology freak like me.

In today's review of the Spectrum, which I have simply called "Spectrum Inside" in reference to the infamous Intel logo (Intel may forgive me for the misappropriation), I will give a general overview of the Spectrum (here I will limit myself to the standard system of the Spectrum 48 K) and talk a little about the "inner workings". Specifically, I will devote a little time to the most important system component, the operating system in the Spectrum ROM. Where appropriate, I will also make comparisons with the ZX 81 in order to try to make some things clearer, as far as this is possible for me with my layman's knowledge. I am sure that many of us are familiar with the following in some way or other and that there have already been reports about it in this or a similar way, but I would like to give a brief summary of more than 40 years of Spectrum history, which we still look back on from time to time.

But first things first. Just like the ZX81, the Spectrum had the magic component of a ULA that had to meet customer-specific requirements. This ULA concentrated all the required functions that went beyond the processor and RAM and ROM. The Zilog Z80 processor used in the system was capable of addressing up to 64 KBytes of memory with its 8-bit wide bus. These were addressed via the existing 16 address lines. In the following, we will concentrate on the Spectrum, as I myself have no experience with the ZX 81 and therefore prefer to leave the field to the specialists to write profoundly about it.

As already mentioned, the Spectrum was largely based on its immediate predecessor, the ZX 81. All in all, the reduction of components in the Spectrum is not as pronounced as in the ZX 81. Nevertheless, the programming still shows its relationship. In terms of hardware, it is of course a little more elaborately designed, as it now also delivers a colour image, which was still denied to the ZX 81. The best known and most widespread version of the Spectrum was the so-called 48K,

which, as the name suggests, was sold with an immensely large main memory of 48 KByte. If you add the 16 Kbytes for the ROM, then the entire address range of 64 Kbytes that could be addressed by the Z80 was fully utilised. If you now subtract the approx. 7 Kbytes of screen memory from the approx. 48 Kbytes of RAM, you arrive at the known approx. 42 Kbytes of really freely available memory, which you could also use via the integrated BASIC interpreter. You can query all this with the well-known command PRINT 65535 - USR 7962.

As is well known, the Z80 processor has special commands (I already reported on programming the Z80 here in the club newspaper many years ago) and its own control line for input/output operations. This has the advantage that additionally connected devices such as the keyboard etc. do not burden the memory area, because the IN and OUT commands allow the entire 64 KByte to be used as I/O addresses or ports.

Due to the large memory area for that time (we are talking about the beginning of the 1980s) (other home computers, as they were also called, did not have more to offer), it was possible for the first time with the Spectrum to define a fixed screen memory in the system. You won't find a character-oriented screen memory in the Spectrum, although some of us might think so, because characters are visible on the screen. The character set in ROM is responsible for giving the screen content a graphic form. In its day, this was also an essential difference to other computers. As is well known, the screen is constructed in pixels, each dot is a bit in the memory.

What does this mean in concrete terms? I would like to try to illustrate it with a very simple example. When writing a character on the screen (resulting from any keystroke on our part or an output of a program), the dot-like form of the character to be displayed is taken from the character set and stored in the memory as an image, so to speak. Conversely, this also means that the representation of characters and individual graphic dots can be handled identically in the memory. This is why there are no different screen modes for graphics and character sets. If we now calculate a little, we come to the following results. The standard screen of the Spectrum consists of 22 lines of 32 characters each. As we know, each character is mapped from an 8*8 matrix, so in pixel format we have a screen resolution of 22*8 dots per row to 32*8 dots per column or the well-known 176*256 pixels of the screen. As we know, the Spectrum also allows an OVER command, which makes it possible to overlay characters without deleting the character that was previously in that position. In addition, the Spectrum character set also uses so-called graphic symbols and thus also differs from other computers not only of its time. And as the climax of the confusion, there are the good old attribute values that appear in four forms (permanent, transparent, temporary and temporarily transparent). Anyone who can still keep track of this is truly to be envied. I, for one, was totally annoyed for a long time, but eventually, with some patience, I understood it. But enough of that.

Now let's get to what really makes a computer - the operating system. Apart from the hardware-dependent conditions, it is the operating system that enables us to really interact with the computer. Let's clarify once again: the operating system is not the language in which we communicate with the computer (even if it could easily be misunderstood that way with the Spectrum), but the part of the internal software that connects and manages the existing units of the system. These units include, above all, the keyboard, storage media, screen control, printers, etc. As a rule, the actual operating system is largely separated from other memory areas. This also applies to BASIC computers like the Spectrum. Our Spectrum is no exception in that the operating system and the language (BASIC) interpreter are not located separately next to each other in the ROM. As is well known, the Spectrum uses a variant of this language that is based on Microsoft BASIC, although there are some differences, especially in string processing, which I will not go into here.

Now let's take a closer look at a few units and their connection to the Spectrum ROM. Let's start with the keyboard.

If you have a disassembled ROM listing of the Spectrum Standard ROM, you will see that the keyboard is integrated from address **$028e HEX**. Based on the ZX 81, a real masterstroke of effort minimisation was used here. What is meant is the **command 029 ED 78 NXTREI IN A,(C).** This command outputs the B and C registers of the Z80 as addresses. This is equivalent to the row and column of the matrix of the key that is currently being addressed. This does nothing other than decode the keystroke accordingly. Those who have already studied the individual ROMs of the Sinclair computers will notice that the routine is almost identical to that of the ZX 81, but the internal coding differs in essential points. The Spectrum uses a standardised ASCII character table, which also takes upper and lower case letters into account. This is all the more remarkable because there was no trace of standardisation on the ZX 81. But things only get really interesting on our computer when a key is pressed.

Why? Well, the reaction to a keystroke is a really interesting area that does not always open up to our logic straight away. A really complicated system of logical queries does not always result in one letter per

keystroke, because the keys on the Spectrum are, as we know, assigned several times. In addition, all BASIC commands/functions are entered in one-key mode by means of so-called tokens. If you consider that depending on the syntax check and the two differently acting shift keys, up to six possible reactions and thus functions result from one keystroke, this is not without its difficulties. Anyone who has never worked on a Spectrum (for a long time) will often be put to the test by the tedious task of learning the keyboard modes. Once you know how to do it, however, you will be very fast and effective. The BASIC commands can be found in the memory area **$0095** to **$0204**. Analogous to Microsoft BASIC, the last character of a name is marked with $80. This serves to recognise the end of the word correctly. Afterwards in the ROM, the assignment of the keyboard matrix to the ASCII numerical values is realised. Through the evaluation in the context of the syntax check, it is then recognised whether it is a character or a BASIC token. These tokens of the corresponding functions, which thus represent 1-byte abbreviations, are stored in groups in the range **$022C** to **$028D**.

The main storage medium in the standard version of the Spectrum was usually the cassette recorder. The extensive routines that enable the control of this storage medium are stored in the area from **$04C2** to **$09F3**. The ZX 81 had the special feature that the connection of the video signal and that of the recorder output were identical.

This has changed with the Spectrum. Similarly, the recorder output and the colour output of the screen border have been linked in the Spectrum. This makes it easier for experienced users to recognise the temporary states when working with the storage medium. Technical errors can also sometimes be deciphered from the stripes of the screen border and can possibly be used for troubleshooting in case of storage problems. However, and I can say this from my own experience, you have to be very familiar with the subject matter.

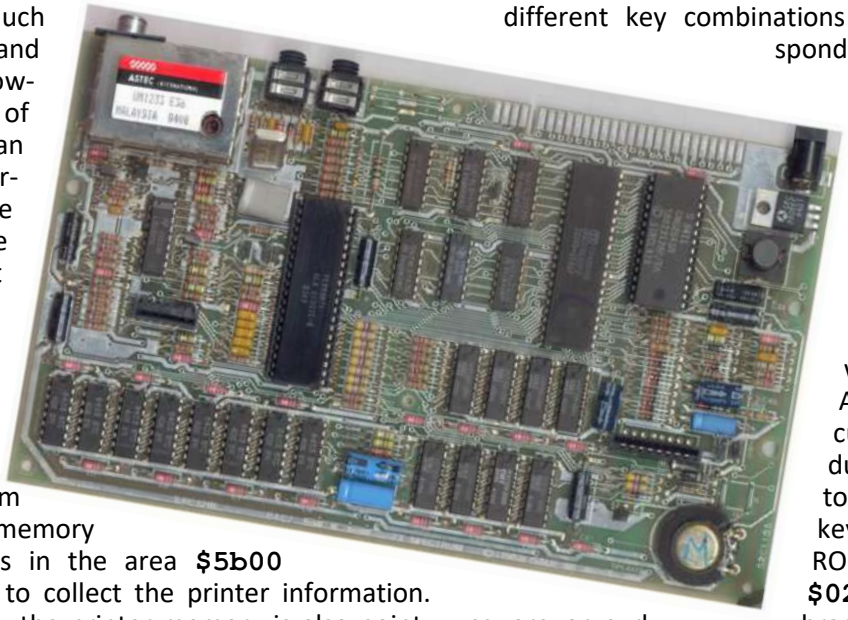
Significant changes have been made to the recorder routine compared to the ZX 81 with regard to the header. In addition, reliability when working with the recorder has been improved considerably compared to the previous model. In particular, the stable screen output of the Spectrum is a great help here. A new addition is the **MERGE** Routine, which allows the appending of program parts and variables. So much for this topic.

Let's now turn to the sound routines and the sound generator. In the memory area **$0385** to **$04A9** there are some routines on this subject. These serve to control the built-in loudspeaker (even though it really doesn't deserve this name). It is rather a beeper, as the English name suggests. The connection itself is more than simple. The beeper is connected to the output port **$FE bit 4**. If a program or a command outputs a zero there, the current flow is switched on, a one then switches it off again. As almost everyone knows by now, the Spectrum, unlike some other 8-bit computers, does not have a sound controller. Therefore, the Spectrum's processor has to take over and create a sound in the right rhythm by switching it on and off. Since it is not easy to produce a reasonably pure sound with this method, the times of switching on and off have to be kept exactly. With this, you can perhaps imagine what happens if the sound generation is interrupted by a program interrupt, for example. Therefore, during sound generation, the interrupts must necessarily be disabled. This also means, however, that the keyboard, for example, cannot be addressed for a short time. The Spectrum obtains the values of the period duration by means of internal calculation routines by converting the desired time duration and the frequency into period duration or counting pulses and number of periods. So you see, it can be a little complicated.

What would a computer be without the output on the screen? I already mentioned above that the Spectrum has a fixed screen memory that reserves one bit per pixel. This screen memory starts at $4000 (16384 - sounds familiar?) and goes up to $5800 (22528). These are then the famous 6144 bits of a screen without attribute values. This corresponds to 192 lines with 256 points each, provided that 8 points fit into one byte. If you have been paying attention, you may have noticed that 192 lines correspond to 24 lines of text? This is because the Spectrum has a split screen area. Namely, the well-known 22 lines of the upper area and the 2 lines of the lower area. Under normal circumstances, the user can only access the 22 upper lines within a BASIC program. These are then again the 176 vertical pixel positions already mentioned above. The two lower text lines are used for inputs and system messages. However, the lower area can change dynamically if a correspondingly large amount of text is entered. This also means that BASIC lines can grow beyond the normal width of a line. With the Spectrum, one might have wished for a better editor, like on the Commodore or Epson computers. Those who know the good old Apple II, on the other hand, are quickly satisfied with the editing possibilities of the Spectrum. A previously selected line (see cursor sign >) is made available for editing by means of the EDIT command. You can now make entries at any point, but existing characters are not overwritten or deleted, but rather moved forward. This is sometimes quite helpful if you don't want to lose everything that is already there. On the other hand, you still have to remove superfluous characters with DELETE before the line can be passed to the interpreter for syntax checking. This editing function originates from the ZX 81. As is well known, the screen memory is pixel-oriented, i.e. everything that can be seen is represented by individual dots. This makes it possible to display characters over characters or to draw lines through texts. All this is no problem. Graphic figures and texts are simply identical for the Spectrum.

We have already briefly mentioned the attributes above. They follow the memory area of the points and have a length of 768 bytes. This means that for each character field in the size of 8*8 bits, there is a corresponding colour indicator. That is 24*32 fields for the whole screen. This means that with the Spectrum you cannot really speak of the output of a high-resolution graphic. This is also because colour information can only be defined per character field and not per character.

A little calculation example? If we really wanted to display eight colours per screen dot, we would need 3 bits per dot. Thus we would need 3*6144 bytes = 18432 bytes. This would make the screen memory three times as big as it is now and we would hardly have any more space for the program. But if you consider that even the current way requires almost 7 Kbytes just for the screen, you quickly come away from such demands. While the ZX 81 uses sophisticated software to generate images (interrupt-controlled line layout!!), the Spectrum uses hardware. This is only possible because the Spectrum has a much higher clock rate and thus computing power. The generation of images is only an unnoticeable burden on the Z80, the management of the screen content costs it a little more.



Next to the screen, the printer is probably the most widely used medium for output. A memory page of 256 bytes in the area **$5b00 (23296)** is used to collect the printer information. With the Spectrum, the printer memory is also point-oriented and therefore graphics and text can be mixed without further ado. The printer does not need any colour information. I will return to printing issues in more detail later on.

But what is the first thing the user encounters after the system "boots up"? We are not talking about the status message with which the Spectrum announces itself as ready for use, no, we are talking about the BASIC interpreter which is already active at this point and which is waiting for the user's command input. The interpreter integrated in the ROM performs the tasks of line interpretation, direct command execution and the calculation of mathematical expressions, for which it contains the arithmetic routines and the existing floating point calculator, which, for example, performs polynomial calculations of the trigonometric functions. In contrast to the ZX 81, the Spectrum has an extended range of these functions and their inverses. The instruction set has been noticeably extended compared to the ZX 81. Another problem is logical commands such as **AND** and **OR**. They are not easy to represent for almost all interpreters, since they react fundamentally differently and are not at all suitable for logical checks within a byte or within decimal numbers up to 255.

If one were to take into account the handling of bits within bytes, this would have to be done through an enormously complex dividing and checking process. The well-known graphic commands **PLOT**, **DRAW** and **CIRCLE** are typical for Spectrum. As is certainly known, **CIRCLE** uses very small straight line segments when drawing arcs. This is mainly for performance reasons.

Anyone who has never worked with a Sinclair computer, and in particular with the Spectrum, will despair at the sometimes very unconventional up to sixfold key assignment. I have already mentioned this above. Admittedly, it takes time to get used to the individual modes, but once you have got to grips with them, you don't want to do without them. At the time, the background to this was that it was absolutely necessary to map all BASIC commands using a so-called "one keystroke".

Considering the different shift modes that have to be mastered by the key to be pressed, pressing the different key combinations to get to the corresponding mode often increases the number of keys to be pressed beyond the letters that the final command to be reached actually has. I hope the sentence was not too convoluted and you could understand what I wanted to say. As an example of the curiosity of this procedure, here are the red tokens under the letter keys (to be found in the ROM from address **$0249**). To reach the square or curly brackets, i.e. a single character, one must first press **Symbol Shift and** Caps **Shift** to enter the **E-mode**. Then continue by pressing Symbol Shift together with the key for the character you wanted. Admittedly, this is a little grotesque.

Further on in the ROM, you get to the displayable characters (text and graphics) in the last three memory pages, i.e. from **$3D00** to **$3FFF**. As mentioned above, the characters consist of an 8*8 matrix and therefore occupy a lot of space. Those who have counted will notice that the Spectrum "knows" a total of 96 characters. It should be noted at this point that the rough graphic characters in the "quarter design" are not in ROM, but are calculated. What is unusual about the character set is that the characters leave a dot position at all ends or rather at the edges. This is not found on other computers. When the characters are printed on the screen, they are not spaced apart but border directly on each other. For this reason, the characters in the matrix do not go all the way to the edge, as they would otherwise run into each other on the screen. However, this measure does not necessarily contribute to the clarity of the character set, because the matrix per character is reduced from 8*8 to 6*6 pixels. This is why the character set of the Spectrum is not necessarily known to be very precise and easy to read.

As already mentioned above, we will now briefly deal with the topic of peripheral devices to be connected and the resulting I/O problems. However, only the software side will be considered here. Everything else is a matter for the hardware specialists, and I am certainly not one of them. As is well known, the keyboard

also contains commands for a rudimentary storage system in the form of external "floppies". The commands **CAT**, **ERASE**, **FORMAT** and **MOVE** simply run to an **"INVALID STREAM"** in ROM at memory location $1793. The control software is not present in the Spectrum and must be "brought along" by the external storage device itself (in its own ROM). There have already been many interesting articles about this, especially about the available diskette systems and their handling. At this point, we would like to refer to the corresponding technical articles here in our club magazine (if you want to know more about what is available on the individual topics, please refer to the "Index" of the club articles or the internet search). You can find almost anything on the net.

The Spectrum has the commands **IN** and **OUT** for addressing input and output extensions. These are completely identical to the Z80 code (see R. Zaks). The output takes place at memory location $1E7D in the form of **OUT (C),A**, the input at memory location **$34A8** with **IN A,(C)**. If you want to know more about this topic, we recommend reading the manual or the literature about the Z80 code.

The Spectrum uses very special ports for its individual system units such as the keyboard, beeper, recorder connection or printer. If the user knows these, he can also operate his own external devices from **BASIC**.

Dear fellow users, I would like to end this short review under the "rubber mat". When you have been working with the Spectrum for many years, it is always interesting to go back to its roots and recall some things.

As we have seen, the Spectrum is a strange fellow, especially when you consider the special features described. But perhaps that is also the reason why working with it is so much fun. I apologise for any undiscovered mistakes, they were not intentional.

© Harald Lack

HOW THE IMAGE GETS ONTO THE SCREEN

BY CHRISTOF ODENTHAL

Actually, I wanted to write mainly about Tangerine's ORIC and its "crazy" image composition in this article, but now it turned out to be a little more extensive.

I won't go into the history of the ORIC here, you can read about that in SUCSession No. 227. Only this much: The ORIC was intended as a competitor for the ZX Spectrum, but could never reach this status. However, I would like to say that I consider the ORIC Atmos with its black and red case design and its "real" keyboard to be one of the most beautiful (if not the best) 8bit computers of the era.

The main reason for this article (which is mainly about the graphics display) was this very relationship and the tricky construction of the image memory, which allows for a larger number of different colours in a smaller area under roughly the same conditions - but you have to accept other limitations for this.

Through my collection of home computers, I know many systems and their image memory structure, especially when it comes to high-resolution graphics. The reasons for the sometimes complex setups are mainly memory savings (RAM memory was very expensive at that time) and technical reasons, i.e. a simplification of the hardware and thus also a cost reduction. Most systems also have to live with a brake on the processor while the graphics chip accesses the image memory. Sometimes this only affects certain memory areas, when the graphics chip cannot access the entire memory - this is then usually called "contention memory" or "Chip RAM" (as opposed to "Fast RAM"). With the MSX 1 there is no brake because the video chip has its own memory - however, it cannot be accessed by direct memory access but only by OUT command, which slows down the access again!

SYSTEMS WITH SEPARATE GRAPHICS AND COLOUR MEMORY

In these systems (e.g. ZX Spectrum, C64, MSX 1 Mode 2) the graphics are monochrome (each bit a pixel) and the colour is determined by the separate colour memory (via so-called "colour attributes") - usually for an area of 8x8 (ZX Spectrum, C64) or 8x1 (MSX 1) pixels.

SYSTEMS WITH COMMON GRAPHICS AND COLOUR MEMORY

In these systems (e.g. Amstrad/Schneider CPC, Atari, MSX 1 Mode 3), several bits of a byte determine the colour for the corresponding pixel.

The colour palette itself is either fixed (ZX Spectrum, C64, MSX 1, ORIC) or can be selected from a larger colour space (CPC, Atari).

MIXED FORMS AND VARIANTS

In the C64 hires multicolour mode, two bits of a byte are combined and refer either to one of three colour attribute memories or to a register with a common background colour. In this way, four different colours can be displayed in an 8x8 pixel area, but the resolution is reduced by half.

With the ORIC, bytes with monochrome graphics and colour attributes can alternate in the same memory, i.e. the attributes are not in a separate memory, as for example with the C64 or ZX Spectrum.

With the MSX 1, I would like to mention mode 1, which is a text mode (with a freely definable character set), but has an unusual colour scheme: In the character set, eight consecutive characters can be assigned a common colour, i.e. the characters with

Code 07 can have one colour, the characters with Code 815 another, etc.

STRUCTURE OF THE GRAPHIC MEMORY

There are also differences in the structure of the graphics memory. You might think that the pixels you see on the screen are stored (in the form of bytes) in the same way in the memory, i.e. line by line from top

left to bottom right. But this is not the case! You can observe this if you fill the image memory linearly with bytes while looking at the screen - you have probably noticed this before when loading a screen! The following program (on the ZX Spectrum) also illustrates this - slowly to watch:

```
 5 CLS
10 FOR n=16384 TO 22527
20 POKE n,255
30 NEXT n
```

When I talk about lines in the following, I mean pixel lines, not character lines - one character line is usually eight pixel lines!

There are the following methods (and combinations of them):

- Simplest case: the bytes with the pixels are read and displayed line by line (Atari).
- Interlace method: This also starts at the top left in line 1. The bytes/pixels then follow directly next to each other until the end of the line is reached. The next line is not line 2, but a few lines are skipped (on the ZX Spectrum and CPC, one line has eight pixels between it and the next). At the end of the screen (or block, see below), line 2 follows and so on.
- With the C64 and MSX 1, you also start at the top left and first fill an area the size of a character, i.e. 8x8 pixels or eight bytes - that is then 8 lines one below the other with a width of one byte. Then, again starting at the height of the first line, the next 8x8 pixels follow to the right. When you reach the edge of the screen, you continue eight pixel lines lower on the left.
- The screen can sometimes also be divided into blocks - a block goes from the left to the right edge of the screen, but does not fill the whole screen towards the bottom, but only a part of it (on the ZX Spectrum and MSX 1 Mode 2 there are three vertical blocks). When filling the screen memory, one block is filled after the other, according to one of the above procedures, before it is the next block's turn.

The *Amiga*, by the way, does it quite differently, it has so-called »bitplanes« - these are basically separate graphics memories that are placed on top of each other. Each additional bit plane adds a colour bit, i.e. with only one bit plane the graphic is monochrome, with two bit planes you can display four colours, with three eight and so on. The disadvantage is that you always have to address / write to several different memories at the same time to get a multi-coloured graphic on the screen. To compensate for this, the *Amiga* has a special chip called »Blitter« to move large amounts of data, but it is still a disadvantage, for example when fast graphics are needed for 3D-Ego-Shooter. By the way, the *Amiga* CD32 has another (exclusive) special chip called *»Akiko«* that can do *»chunky to planar«* conversion (a distribution of colour bits from a common memory "chunky" to several bitplanes "planar") - but the chip was mainly intended for fast display of videos from CD and is rarely used.

The big disadvantage of these sometimes quite complicated image memories is that you need equally complex algorithms to write a rectangular area into them, e.g. a sprite or background graphic - and that makes the programs slower!

THE GRAPHIC CAPABILITIES OF THE »ORIC«

After the above excursion, let's move on to the *ORIC!* It can display 8 different colours, has a text mode (»TEXT«, »LORES 0« or »LORES 1«) with 40x28 characters (where one character consists of 6x8 pixels) and a graphic mode (»HIRES«) with up to 240x224 pixels. The special feature is that although a line on the screen consists of 40 bytes - which would suggest a resolution of 40 bytes x 8 bits per byte = 320 pixels - the resolution is reduced to 240 pixels by using only six bits per byte. The remaining two bits are used to control the image output. The difference between the two modes is that the image memory in TextMode refers to characters from the character set, whereas in GraphicsMode it contains the pixels directly. TextMode uses 3040 bytes of memory (including character set) and GraphicsMode uses 10210 bytes (with three lines of text and character set) - graphics-only memory would be 8960 bytes. In comparison: The image memory from the *ZX Spectrum* is 6912 bytes.

THE ATTRIBUTES OF THE ORIC

As mentioned above, the ORIC has no separate attribute memory. Graphics and attributes share the same memory - but this also means that where there is an attribute, there can be no character and no graphic! So if you want to change the colour for the following characters or pixels, you have to live with a six pixel wide gap in foreground or background colour.

The attributes are called »serial attributes« because their effect ends at the end of the line at the latest (with the exception of the video control attributes - they remain until they are changed again). Each time a new line is started, the background colour is reset to black and the foreground colour to white. An attribute is marked by bits 6 and 7 being zero - a byte in the image memory with a value of 031 = bits 15 (or 128159 if bit 8 is also set) thus represents an attribute. Values between 32127 are either a character or pixel depending on the mode (bit 16 = pixel, the 7th bit must be set). Bit 8 inverts the colour of the current character, the six pixels or the current attribute. Unfortunately, the ORIC does not have a "Bright" attribute, i.e. its eight colours are only available in one brightness. The border is also always black and cannot be changed.

ATTRIBUTES FOR VIDEO CONTROL

But there are not only attributes to change the foreground or background colour, you can also make them blink (as with the *ZX Spectrum* - although in contrast only the foreground colour blinks, not the background colour). However, the attributes can also be used to switch between two character sets (128 characters each) in text mode or to switch the display of the char-

acters to double height (i.e. 6x16 pixels) - the upper half of the character is then visible in every even line and the lower half in every odd line!

Even the text and graphics mode is controlled via the attributes - and that with either 50 or 60 hertz refresh rate! So you can mix graphics and text, but you should choose one of the two frequencies, otherwise a video signal could be generated that the TV / monitor cannot display!

Attribute	Value	Foreground-Colour	AttributeValue
0	Black	16	Black
1	Red	17	Red
2	Green	18	Green
3	Yellow	19	Yellow
4	Blue	20	Blue
5	Magenta	21	Magenta
6	Cyan	22	Cyan
7	White	23	White

Attribute Value	Effect	Attribute Value	Video Mode
8	Standard Character Set	24	Text Mode 60 Hz
9	Alternative Character	Set	
10	Double Standard	26	Hires Mode 60 Hz
11	Double Alternative	27	
12	Flashing Standard	28	Text Mode 60 Hz
13	Flashing Alternative	29	
14	Double Flashing Std.	30	Hires Mode 60 Hz
15	Double Flashing Alt.	31	

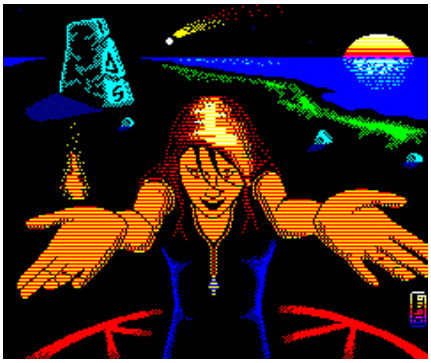
Table : Attribute values

A FEW TRICKS

As with the ZX Spectrum, there are a few tricks to hide the limitations of the graphics a little. At the beginning of each new line the colours are automatically reset to black (background) and white (foreground) - usually you set your own colour attributes for the following graphic, for which you need two bytes (one each for background and foreground), thus 2x6 = 12 pixels are lost. Since the border is always black, a trick is to set the attribute for the foreground colour first in the first byte of the line and only in the second byte the attribute for the background colour - this way it looks as if the border is six pixels wider (because the background is still black for these first six pixels) and you seem to lose only six pixels (in the second byte) for which you cannot set a graphic because the foreground colour attribute is set there. But you can also just not set any attribute at all and just leave the colours black / white to use the full 240x224 pixels for a monochrome image.

»Beast«, by Twilighte (Source: osdk.org)

With *the ZX Spectrum* you can (without tricks) only select two colours for each 8x8 pixel area. With the ORIC, this is possible for every 6x1 pixel area - so you can set new colours in every pixel line. If you now change the colours in every second line, the colours blur into a new one from a greater distance to the screen.



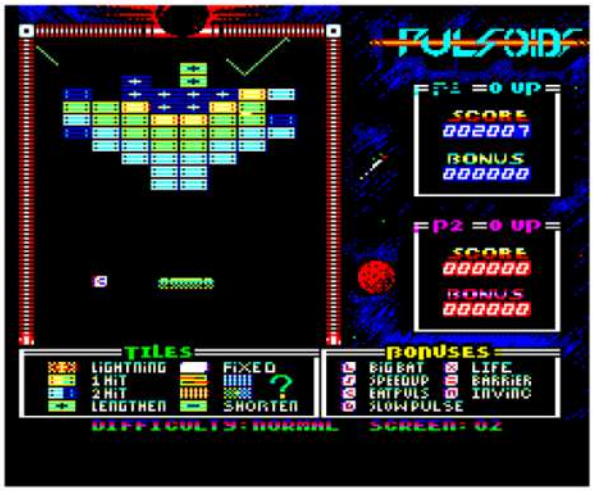
»Yessagician«, by Dbug (Source: osdk.org)

Another trick is the clever use of the colour inversion bit (bit 8, value 128) - this can be used in any attribute, character or graphic byte and immediately changes the colours (using the following table) without having to use a colour attribute and thus give away 6 pixels.

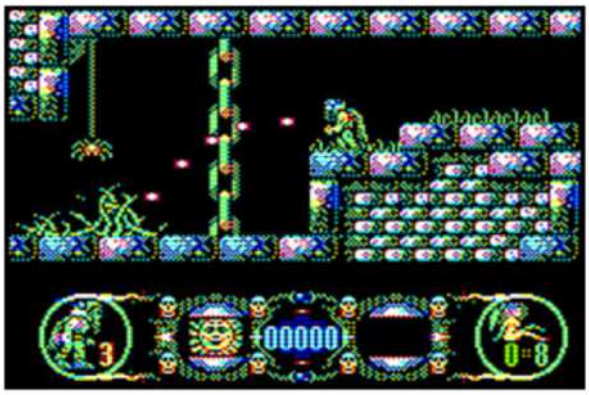
Original colour	Inverted colour
Black	White
Red	Cyan
Green	Magenta
Yellow	Blue
Blue	Yellow
Magenta	Green
Cyan	Red
White	Black

Table: Inverted colour values

»Ghost Gobbler«, (Quelle: osdk.org)



»Pulsoids«, by Twilighte (Quelle: osdk.org



»Stormlord«, by Twilighte (Quelle: osdk.org)

By the way, most games for the ORIC are programmed in text mode (just like on the C64). This allows fast animations - simply by changing a character in the character set, because then all the places on the screen where this character is used change abruptly - and this without time-consuming copying by the processor - the graphics chip does this for free, so to speak!

Finally, I do not want to conceal another major disadvantage of the ORIC (besides a missing BRIGHT attribute): There is no interrupt (signalling) when a new video image is built up - this is the only way to synchronise e.g. animated graphics with the image build-up to avoid flickering. However, there is a hardware modification with the cassette interface to retrofit this signal. The usual problem is that unfortunately not everyone has this modification, so that one cannot assume that it is available.

»THE ZXPICOMD«

ANOTHER ALTERNATIVE TO THE MICRODRIVE/IF1

BY HEINZ JUNEK, 08 JANUARY 2023

Another alternative to the Microdrive for the IF1: The ZXPicoMD modern alternative to the unreliable Microdrive with all its advantages and more.

In issue 233, p. 18, I wrote an article about the *Oqtadrive* as an alternative to the microdrive for the *IF1* with the classic "IF1-feeling" but 100% reliability. This device was developed jointly by xelalex and TomD.

At the same time, `https://github.com/TomDDG/ZXPicoMD` has also released its own product *»ZXPicoMD«.*

Technically, both devices are identical in terms of communication between the device and the IF1. The main difference between the two devices is, in my opinion, that xelalex places emphasis on WLAN communication with the PC, while TomD places more emphasis on the independent "existence" of the *ZXPicoMD*. This means in particular that the archive of mdr files is largely managed on the SD card with the *ZXPicoMD*, while *xelalex* does this mainly on the PC.

The *ZXPicoMD* has its own small OLED display, which can be used to control the unit independently of the IF1. The *ZXPicoMD* is operated (in addition to the *IF1* commands) by means of five push buttons. Various functions can be selected here: Loading and saving cartridges from and to the archive, displaying cartridge contents (see illustration) and formatting. Real Sinclair microdrives can also be connected and inserted at any point in the drive chain via the push buttons. The name "*ZXPicoMD*" comes from the use of the *Raspberry Pico*.

Since this is much cheaper than a *Raspberry Pi Zero* and an *Arduino nano* is not needed at all, the construction costs are much lower. Including the housing, they are around 20€. By using the *Raspi Pico*, the device is ready to use in less than a second, and it also doesn't need a slow shutdown before switching off the ZX Spectrum.-These two processes take about 30 seconds with the *Oqtadrive*. The installation of the *ZXPicoMD* also does not require LINUX access, which lowers the threshold for a replica. In the future, communication with the PC via WLAN is also planned. When rebuilding, it is therefore recommended to use a "*Raspberry Pico* W". All in all, this new device is at least as much fun as the *Oqtadrive*.

»DIZZY - 4 MAGICLAND«

GAME DESCRIPTION AND REVIEW BY HARALD LACK

R. Fred Williams, Peter J. Ranson, Chris Graham, David Whittaker and Neal Vincent
(1990 Code Masters Ltd)
Hardware used: Original ZX Spectrum 48k

Hello friends of Dizzy games!!!

Today's post is not about a (text)adventure solution this time but the solutions to classic Dizzy games with Dizzy 4 Magicland Dizzy. The game was released in 1990 and was distributed by Code Masters Ltd. R. Fred Williams, Peter J. Ranson, Chris Graham, David Whittaker and Neal Vincent were responsible for the programming. From here on, none of the Oliver Twins were on board, although the story was of course based on their characters, which in my opinion was not detrimental to the quality. The whole thing can be played flawlessly on a Spectrum 48K, but it also ran on the bigger brothers. The gameplay has not changed much from the predecessors of this series, there are still a lot of puzzles to solve, you have to run around a lot, handle the objects you find in the right place and try to advance. You will get many hints if you have a "chat" with the characters you meet. But please pay attention to the right "conversation partners". Our task in this game is to free our enchanted friends from the power of the evil Zaks and finally to defeat Zaks himself. But that's not all. Afterwards, we should try to disappear from Magicland and return to our familiar home. However, we can only do this once we have made a deal with the devil, because only he can teleport us back. In return, however, he demands that we hand over 30 diamonds, which we must have collected beforehand. A few more notes on this. The majority of the diamonds (27 to be exact) are directly visible and as a player you tend to simply collect them. However, since we are always threatened by dangers of various kinds in this program (more on this later), which cost us energy and we only have three lives available anyway, we should only pick up the diamonds gradually, i.e. very systematically, because they bring back energy and can sometimes decisively prolong our lives. Of course, we must not forget to collect them all at the end. Those who have been paying attention will have noticed that three diamonds are still missing; these are hidden behind various things and are therefore difficult to find. But I have listed below where you can find all the diamonds. What each item does and what you have to do with them can be seen in the following list:

- Backdoor key -> opens the back entrance to the lock
- A beg of rubbish -> absolutely useless
- Empty milk bottle -> absolutely useless
- The gold cross -> helps against vampires
- Power pill -> removes the ghosts above the water
- The handle -> pulls up the bucket at the well
- The eatme cake -> absolutely useless
- A poisoned apple -> take it to the witch Glenda
- A bucket -> fill it with hot water from the geyser and then take it to Denzil to melt it from the ice
- A cloth duster -> rub it on Dozy's lamp to call Zaks
- A big stick -> hit the goat with it
- A black cat -> take it to the witch Glenda
- A dagger -> use it to cut the goat loose and cut a leaf at Dylan's place
- A leaf -> take it to the witch Glenda
- The goat -> hit the goat with the stick to push the troll off the bridge
- Dylan -> bring the potion from Glenda
- Denzil -> melt with the bucket of hot water
- we get the personal stereo
- Prince charming -> bring the frog Dora
- Witch Glenda -> bring black cat, leaf, poisoned apple and torch
- Dozy -> bring lighning rod, lamp and duster¬ and then call Zaks
- Troll -> drive away with the goat
- The Bard -> bring the personal stereo
- The drinkme potion -> bring to Daisy to shrink her
- Dora the frog -> bring to Prince charming
- A personal stereo -> bring to the Bard we get the magic pipes
- Some magic pipes -> take them to the rat in the well and give him a run for his money
- The Rat -> drive him away with the magic pipes
- Something sticky -> use it to pull the sword out of the rock
- An ancient lamp -> rub it with the cloth
- Ghosts -> drive them away with the power pill

- A lightning rod -> bring it to Dylan
- Lever -> enables a way across the water
- A rail -> hide diamonds
- Vampire -> expel with the gold cross
- Monkey -> avoid the nuts
- Excalibur -> with it you become king and can beat the queen in chess
- The torch -> opens the entrance to Daisy then take it to the witch Glenda
- Daisy v give her the drinkme potion
- weedkiller potion -> bring it to Dylan
- Devil -> from him you get the wizard slayer trident.
- Wizard slayer trident -> use it to stab Zaks
- Zaks -> stab with trident you get the ring of Zaks
- Zaks ring -> place it in the screen left of the devil on the middle rock
- Mosquitoes -> be careful, they cost energy

So much for the tips and hints. I hope I haven't forgotten anything. The graphics are once again more than well done and it is of course fun to explore the surroundings. Of course, there are one or two small graphic clashes, but that doesn't detract from the gameplay. You're used to that from the Spectrum. Just like its predecessors, Magicland Dizzy is not a game that you just play through. You can occupy yourself with it for quite some time, although it does tie you to the Spectrum - at least that's how it was for me. To make sure that life is not made too easy for us, there are also some things and creatures that want to get at us. It's particularly tricky that sometimes you have to position your character with pixel precision to jump off and occasionally you simply have to jump into the unknown. But of course, that also makes for a certain charm. If you would like to have a look at our "field of activity" in its entirety, I have added an address at the end where you can view an overall plan that really shows how large the playing field is. Now we know what we have to watch out for. Now, for all those who don't want to spend a long time looking for the diamonds, a collected list of the places where they were found:

West Tower	Behind a railing in Sword in the Stone
Zaks Tower	Throne Room
Curiouser and Curriouser	Castle Back Door
Chess Board	Behind a railing in Bushy Grove
Main Hall	Mysterious Monolith
Entrance Hall	Weirdhenge
Crypt	Haunted Swamp
Oubliette	Under the Cauldron of Glenda
Troll Bridge	Hot Water Geyser
Top Watch Tower	Hell Gate
The Lower Tallest Tower	Hades
Bards Tree House	Cracks of Gehenna
Tower with a View	Behind a Railing in the Cracks of Gehenna
In the Clouds	Down a Well
Up a Tree Secret	Passage

So that's it for those of you in a hurry.

Now we come to the step-by-step solution. As always with Dizzy games, this program can be solved in different ways. Everyone is welcome to put together his or her own solution. There is no single solution. But of course there are some dependencies to consider if you want to get further. In the following, I have presented you with the way I played through (hopefully not forgetting anything or writing anything down incorrectly), which, however, as always, does not claim to be the only true one. Let's start ...

We materialise in Weirdhenge and continue from here as follows:

Since it made sense in the solutions of Dizzy 2 and Dizzy 3 to go left in the first screen, we do the same here and find a key at the foot of the mysterious monolith, which we pick up immediately. Then we go two more screens to the left and come to an old well over which we jump (be careful not to fall in or the game will be over) and continue to the left. We are at the moat where a shark is swimming excitedly back and forth. With the right timing we jump onto its dorsal fin and try to save ourselves by running/jumping to the left to the entrance. We use the key we brought with us and the door opens. We immediately put down the milk bottle we just picked up, as it is of no use (see the list above). We go left and find a crank (handle) which we pick up, then up the staircase on the right in the screen and there again 2 screens to the left. Here we go up and follow the path. Here you will find a power pill which you can take with you and then immediately go back down. Now three screens to the right. There is a window with a bat flying around. We jump onto the window ledge and then onto the cloud, being careful not to touch the bat, as this costs a lot of energy. Then we jump to the right and from this cloud we drop down onto the roof of the old well. From here we jump to the right onto the cloud and then immediately to the left onto the other cloud, picking up the "cloth duster". Then we drop down again and position ourselves to the left of the well. We use the crank we brought with us and pick up the bucket for it. We put down the cloth duster and jump over the fountain. From here we go four screens to the right until we reach the water where we see the ghosts flying back and forth. As we have the power pill with us, we can eliminate the ghosts. We jump to the right onto the floating platforms, but be careful that we can only stand on them for a short time before they sink with us. But please don't panic at this point, with a little skill this is no problem. On the second platform we see Dora, whom we pick up before we jump further to the right and reach a small island on which the witch Glenda can be seen. Since we have eliminated the ghosts and they no longer appear, we can throw away the power pill. We talk to Glenda and learn that she has been looking for her cat for a long time. From here we continue to the right (again over floating platforms) and try to keep the mosquitoes at bay. We continue to the right until we reach the geyser, where we fill the bucket we brought with us with hot water in the small puddle. On the way back we take the "drinkme potion" with us and return to the large hall of the castle. Here we put down everything we have with us, only with Dora we go one screen to the left and give her to

Prince Charming. Then back to the right and up the staircase, to the right again and from here we go back over the clouds to the roof of the fountain. From here¬ we jump twice to the right and get the stick there. Then up again and from the middle of the top right platform jump further to the right and find Glenda's black cat on the monolith. Collect it and at the far right edge jump further to the right and land on top of Weirdhenge. We take the dagger and from here jump twice to the left to get past the torches unscathed. Then we go back to Glenda and give her the cat. Out of gratitude that we have found her pet again, she offers to help us. We learn what she needs to free our bushy friend. Armed with this information, we make our way back to the Great Hall. When we pass Dylan, we cut a leaf off of him, but we leave it for now. Once in the hall, we pick up the bucket deposited there and go left twice. We are on the other side of the moat. There is a lever with which we can switch on a kind of drawbridge. Now we are able to cross the moat. From here we turn left again three times until we meet a goat at the troll bridge. We cut the goat loose with the dagger, but it remains rooted to the spot. So we hit it with the stick and persuade it to run away. On her escape she also pushes the troll off the bridge, which never hurts. We continue to the left twice and here at the entrance to the Ice Castle we find Denzil frozen in a block of ice. We pour the hot water on him and Denzil disappears. Before he does, he leaves us his walkman. With that we go back to the big hall where we put the walkman down. Instead we grab the "drinkme potion" and climb the stairs again. Again we jump onto the window ledge and from here up to the left onto the roof of the bed. On the far left of the bed we now jump to the left and come to a railing. Here we go up and do this for two screens. Here lies the "lightning rod", which we pick up, then nothing but down and to the left. Here we grab the gold cross and continue to the left. Here you can see an open window from which we jump. Turn left and leave the lightning rod with Dozy. From here we go 3 screens to the left and then down twice. Through the cross, the vampire avoids us and we can continue our way. However, it is impossible to avoid the bat flying around there. But if we are not too clumsy, it will only touch us briefly and not cause too much damage. Behind a staircase you can see a torch, which we pick up. This opens a secret passage on the bottom right and we can slip through to Daisy on the right. We use the "drinkme potion" we brought with us and Daisy is freed. Then we go back to the left, up twice and to the right four times. We pick up the apple and put down the cross for it. Then we continue to the right until we reach Dylan and pick up the leaf we cut off earlier. With these three things (torch, apple, leaf) we make our way back to Glenda and give her all three things. Before doing so, however, we have to make sure that we have removed the diamond from under the cauldron, otherwise we won't be able to reach it when the fire is burning. Glenda now brews us a weed-killing potion (what a word) and with that we return to Dylan. We apply the potion and Dylan is also freed. We go to the great hall and pick up the cloth duster on the way. Once there, we put it back and grab the walkman. Now climb the stairs again (stamina is needed in this game) and turn left twice. Jump out of the window again and go down. From the far left edge of the platform we jump to the left, from the middle of the platform we jump to the right and here again to the left. In this way we reach the bard's hut, who is there, unimpressed, blowing away on his flute. We give him the walkman and the guy literally freaks out. He throws away his pan flute and disappears. We pick up the flute and walk up to the fountain. We stand on the right side (important!!!) and drop down the well shaft to the left. We stop in front of an entrance which is blocked by a huge rat. Therefore we jump to the right onto the small ledge and play a little song with the flute. After a while the rat comes to life and attacks us. But since we are standing on the far right, it doesn't reach us and falls into the well shaft. Good riddance. We jump to the left again and continue in this direction until we come to something "sticky" which we take with us. Continue to the left and we find an antique lamp, which we also pick up. Beware of the acid drops falling from the ceiling, you can quickly lose your life. We jump up and put down the pipes. Instead we take the "cloth duster" with us. We go to the sword in the stone and pull it out. A message tells us that we are now King Dizzy. We now go to the left to Dozy and put the lamp near him. Then we use the cloth duster and Zaks appears. Because we have placed the lightning rod directly in front of Dozy, Zaks jumps over directly onto it and frightens Dozy. Therefore he wakes up from his deep sleep and can also go home. Now we have almost freed all our family members, the only thing missing is Grand Dizzy. But we will find him too. We go into the Ice Castle and jump up to the left. From here we jump up twice and here we jump to the left again at the leftmost point of the upper left platform. After this long jump we go down and now see a mirror on the left wall. We jump through it and everything goes the other way round. It takes some getting used to but you can manage. We are on the chessboard and since we are King Dizzy, we can beat the Queen. If we didn't have the sword Excalibur with us, we would be a pawn and the queen would beat us immediately. So we were lucky again. We continue to the left, being careful not to collide with the pawns, which would immediately drain our energy. We free Grand Dizzy, who is standing here a bit awkwardly and doesn't know how he got here. But the guy likes to sleepwalk. From here we now go to the complete opposite end of Magicland, to the volcano crater. After we have rescued all our family members, little clouds rise from the crater and help us to cross it. Continue to the right and then go down twice. In the bottom right corner stands the devil surrounded by a wreath of flames. We try to avoid the flames running back and forth as best we can and step in front of the devil. We get a trident with which we now have to go back to Ice Castle to drive Zaks away. Therefore we have to go all the way up there and here it makes sense to stand as far to the right as possible in the room with Zaks, so that the lightning can't be dangerous for us. At the right moment we jump to the left and stab Zaks with the trident, which makes him disappear. Only his ring remains and we take it. With it we return to the devil and exit the screen at the bottom left. We jump onto the middle rock and throw the ring into the lava on the

left side. Zaks is finally defeated. In the meantime we should have collected all 30 diamonds and can step in front of the devil again. His wreath of fire is now removed and he allows us to teleport home. The price for this, how could it be otherwise, is exactly 30 diamonds. What a coincidence. Well, in any case, we have successfully completed Magicland Dizzy. Hopefully you had as much fun as I did and you made do with the energy and lives you had. I, for one, needed a few attempts.

Finally, a few hints for those who want to "trick" a little. I have taken the following pokes from the internet, but I have not tried them myself, because otherwise I would miss the attraction of the game. If you can trust the sources, some of them are MultifacePokes. Now I would like to list them here quickly for the sake of completeness but without guarantee:

Infinite lives:	29623,182
Run through walls:	41179,201
Speed up the game:	405436,1
Immune to fire:	35903,70
Immune against swamp spirits:	33020,70
Immune against rat:	33085,70
Immune against bats:	32593,70

If you would like to take a look at the overall map, please refer to the Speccy screenshot maps at

https://maps.speccy.cz/map.php?id=Dizzy4&sort=0&part=99&ath=0

here.

And now I wish you lots of fun and luck playing »*Dizzy 4 Magicland*«. The program is definitely worth it.

2022 Harald Lack

KEYBOARD MEMBRANES FOR RETRO COMPUTERS
BY THOMAS EBERLE

Almost 30 years ago, SINTECH's first major investment was the production of keyboard membranes. This was very necessary, as the membranes used by Sinclair in the 80s only had a shelf life of about 10 years and failed by the dozen. As far as we know, SINTECH is the only manufacturer worldwide to produce the keyboard membranes for Sinclair QL and still does so today. SINTECH is not the only one, but with the membranes for ZX Spectrum (rubber) and ZX Spectrum + / 128, they are very popular among repair service providers because they are very accurate, functional and also durable.

Amstrad did not repeat the mistakes of Sinclair and built very good membranes into the later Spectrum units and also into the Amstrad CPC units. Nevertheless, just like the first AMIGA computers, these devices are almost 40 years old, and for various reasons they can still fail. SINTECH has also met this growing need and has produced keyboard membranes for these computers.

The following models are available from SINTECH today:

- ZX81
- ZX Spectrum Rubber
- ZX Spectrum +/128
- Amstrad CPC 6128
- Amstrad CPC Plus 464 / 6128
- Amiga 500 (all models)
- Amiga 600 (green and blue film)
- Amiga 1200

In planning:

- Atari ST
- Amstrad CPC series
- Acorn BBC Micro

Most of the membranes are self-developed. For example, SINTECH developed a universal membrane for the Amiga 500, where previously different membranes were needed.

The development continues. The older the computers get, the greater the demand for keyboard membranes. Initial tests are already underway for Acorn's BBC Micro, but SINTECH is also planning for Atari ST, Amstrad CPC 464/664.

SINTECH notes here that there are of course gaps in demand for keyboard membranes for retro computers. Original quote from Mr. SINTECH: »Of course, we cannot know everything. Are there any keyboard membranes that you miss in our product range? We will be happy to get your feedback and take care of it«.

Mr. Eberle is very grateful for any feedback, as it is his concern to keep the computers on which modern computer technology is based alive and for posterity.

Write to him. He will be happy to receive it:
info@sintech-shop.